Installation

Describes the installation of ioapiTools and additional packages for plotting.Â In order to install CDAT with IOAPI enabled, you need a Fortran compiler.

You can download the source for CDAT at:
http://www–pcmdi.llnl.gov/software–portal/cdat/download

The specific sections are: 1) Express install, 2) seperate python install, 3) Mac fink install, and 4) additonal plotting packages.

---

1. Express install

The easiest way to install ioapiTools and CDAT in general is to do an express install.Â This will build all the necessary software (including its own version of python).Â To do this, go to the cdat top directory, determine where you want to install it, and run:

```
$ ./express_install ~/ --enable-ioapi
```

This will enable the building of the ioapiTools module and all its requirements, and it will build it under your home directory.Â

If you have system administrator privileges and want to install it in a central place on your system, try:

```
$ sudo ./express_install /usr/local/cdat-4.0 --enable-ioapi
```

This will build it under /usr/local/cdat–4.0.

When you want to use it, remember to use the python that you have built through CDAT. So in our home directory case, we would start the interpeter:

```
$ ~/bin/python
```

See CDAT's Installation documentation for the latest information on express install.

Â . contents

---

2. Using separate python install

Instead of building everything, I like to use my distribution's own python.Â The advantage, is that all python modules are in the same, central location.Â The disadvantage is that the build process is a bit harder and more prone to errors.

Prerequisites for building CDAT with your own python is that you need to have the development version of python and the python package for megawidgets.Â In a debian system, get "python–dev" and "python–pmw".

To build CDAT using your own python, you will need to first build the external software and then build

CDAT and the contributed packages.Â Most likely, you will be installing this in a central directory (for example /usr) and will need to be root or have root priveleges through sudo:

```
$ cd exsrc
$ sudo ./install_script /usr --enable-ioapi
```

This will build the external software.Â Check for errors in the logs, especially in ioapi, gdal, and netcdf.

Then build the rest of CDAT and the contrib packages.Â From the top cdat directory:

```
$ sudo /usr/bin/python install.py --enable-ioapi
```

This assumes that your python is the one in /usr/bin.Â If not, you should use the full path to your python.Â Again, check the log files for error, especially pyIoapi.Â

See CDAT's Installation documentation for more details.

Â . contents

3. Mac install via fink

If your using a Mac, you can easily install cdat with IOAPI enabled through the fink package management system.

First you need to have a working version of fink or install it on your system:
http://fink.sourceforge.net/

You need to be running fink in the unstable mode:
http://fink.sourceforge.net/faq/usage–fink.php?phpLang=en#unstable

You can then either install cdat through the program FinkCommander or from the command line:

```
$ fink install cdat
```

This will install cdat with IOAPI enabled and all its dependencies.

Â . Contents

4. Additional packages for plotting

The built in plotting functions in ioapiTools requires the installation of two additional python packages: matplotlib (version 0.74 or greater) and the toolkit basemap (version 0.5 or greater).Â Â Â The matplotlib package is a very flexible and powerful plotting package for scientific plotting.Â The basemap package is an addition to the basic matplotlib package that provides mapping and projection capabilities. These are not needed if you don't need plotting functionality or only want to use CDAT's intrinsic plotting capabilities.

I recommend setting up matplotlib in interactive mode.Â For example, my matplotlibrc file has the following settings:
Â Â Â backendÂ Â Â Â Â : GTKAggÂ Â Â # the default backend

Â Â Â numerixÂ Â Â Â Â : NumericÂ Â # Numeric or numarray
Â Â Â interactiveÂ : TrueÂ Â Â Â Â # see http://matplotlib.sourceforge.net/interactive.htm

I also recommend using the ipython interpreter with the −pylab option to more fully take advantage of matplotlib's interactive mode:

```
$ ipython -pylab
```

Â

See the matplotlib web page for more details on installation, setup, and direct plotting commands: http://matplotlib.sourceforge.net

If your using fink on a Mac, you can install the following packages: matplotlib−py24 and matplotlib−basemap−py24.

Â · Contents